

Fault Isolation with ‘X’ Filter for Bogus Signals and Intensive Scan Cell Sequence Validation

By

KHOR WOOI KIN

**A Dissertation submitted for partial fulfilment of the
requirement for the degree of Master of Science
(Microelectronic Engineering)**

AUGUST 2017

ACKNOWLEDGEMENT

I would first like to thank my thesis supervisor, Dr. Asrulnizam Bin Abd Manaf from School of Electrical and Electronic Engineering of Universiti Sains Malaysia (USM). The door to Dr. Asrulnizam Abd Manaf is always open whenever I ran into torible spot or had a question about my thesis or writing. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the experts who guide me in this thesis including Lim Wei Kheng and Lim Ching Chin, engineers from Intel Microelectronics (M) Sdn Bhd. Without their passionate suggestion and input, this thesis could not have been successfully completed.

Finally, I must express my very profound gratitude to my parents and to my family for providing me with unfailing support and continuous encouragement throughout my study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

FAULT ISOLATION WITH ‘X’ FILTER FOR BOGUS SIGNALS AND INTENSIVE SCAN CELL SEQUENCE VALIDATION

ABSTRACT

There are some concerns in silicon data collection by using the Design-For-Test (DFT). Bogus signal which carries ‘x’ value in simulation, results from the complex logic synthesis and power-up floating state can often mislead the fault isolation process with invalid failing condition. Besides, scan cells within the scan chain architecture is also having mismatch value in between the simulation data and silicon data due to the non-ideal mapping file passed down from the designer team. Hence, it is important to develop an integrated tool that can filter all the bogus signal online and to validate the correlation between silicon data and simulation data with minimum coverage of 90%. Data from actual Intel 6th generation microprocessor with 14 nm process technology, Skylake is imported to ensure that the application of this thesis in the current industry market. Necessary tools such as the “Differentiate and Display” feature to ease the analysis of data, the AND-logic operation to filter the bogus signal and X-OR logic operation to handle the inverted characteristic of signals are developed throughout the thesis. Results show that the developed integrated filter of bogus signals is successful and the minimum coverage of validation tool is 96.5%. Actual failure analysis case from industry is imported and the difference with and without the developed tools are compared. Inconclusive optical test result from the sample is obtained without the implementation of tools. On the other hand, defect of short circuit between the via and the metal line is found after the implementation of the developed tools. It is concluded that this thesis has achieved all the objectives set.

KEGAGALAN PENGASINGAN DENGAN PENAPISAN ‘X’ UNTUK ISYARAT PALSU DAN INTENSIF PENGESAHAN URUTAN SEL SKAN

ABSTRAK

Kebimbangan muncul apabila data silikon dikumpul dengan menggunakan *Design-For-Test* (DFT). Isyarat yang mengandungi nilai ‘x’ dalam simulasi, disebabkan oleh sintesis logik yang rumit dan nilai asal apabila diaktif, sering mengelirukan proses kegagalan pengasingan dengan situasi gagal yang palsu. Selain itu, sel skan dalam rantai skan juga mempunyai nilai yang berbeza antara data silikon dan data simulasi disebabkan oleh fail pemetaan yang bukan ideal dari kumpulan pereka. Oleh itu, pembangunan alat yang bersepadu dan dapat menapis isyarat mengandungi ‘x’ dalam talian serta mengesahkan korelasi antara data silikon dan data simulasi dengan liputan sekurang-kurangnya 90% adalah sangat penting. Data mikropemproses Intel sebenar dari generasi ke- 6 dengan 14 nm teknologi proses diimport untuk memastikan aplikasi tesis ini dalam industri. Alat keperluan seperti “*Differentiate and Display*” fungsi yang memudahkan data analisis, operasi DAN get logik yang menapis isyarat ‘x’ serta XOR get logik yang mengendalikan kebalikan isyarat dicipta dalam tesis ini. Keputusan menunjukkan alat penapisan isyarat ‘x’ yang dibangunkan berjaya dan liputan minima alat pengesahan ialah sekurang-kurangnya 96.50%. Kes pengagagalan keasingan dalam industri diimport dan perbezaan prestasi sama ada dengan alat yang dibangunkan atau tidak dibandingkan. Keputusan ujian optik yang tidak menyakinkan diperolehi apabila alat yang dibangunkan tidak diguna. Manakala, kegagalan litar pintas antara *via* dengan lapisan logam didapati selepas penggunaan alat yang dibangunkan dalam tesis ini. Kesimpulannya, tesis ini telah mencapai semua objektif yang ditentukan.

TABLE OF CONTENTS

Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Problem Statement.....	4
1.3 Research Objectives.....	5
1.4 Research Scope.....	5
1.5 Thesis Outline.....	6
 Chapter 2 Literature Review.....	 7
2.1 The Reasons Behind Failure Analysis.....	8
2.2 DfX Feature.....	9
2.2.1 Build-In-Self-Test (BIST).....	9
2.2.2 Scan Architecture.....	10
2.3 Conventional Fault Isolation and Failure Analysis.....	11
2.3.1 Fault Isolation.....	11
2.3.2 Failure Analysis.....	13
2.4 The Presence of ‘x’ State in Scan Signals.....	14
2.5 Post Silicon Validation.....	16
2.6 Summary of Chapter.....	17
 Chapter 3 Methodology.....	 18
3.1 Setup of Tester and Software Applications in Fault Isolation.....	20
3.2 Generation of Simulation Raw File.....	21

3.2.1 Limitations and Solution in Raw Data Collection.....	22
3.3 ‘Differentiate and Display’ Feature.....	23
3.4 The Development of Algorithm.....	28
3.4.1 Bogus ‘X’ Scan Signal Masking.....	28
3.4.2 Validation on Scan Signal Mapping File.....	29
3.4.3 Usage of “Differentiate and Display” Feature.....	30
3.5 Proof of Concept.....	31
3.5.1 Bogus ‘X’ Scan Signal Masking.....	31
3.5.2 Issue on Possible Over-Filter of Valid Mismatch.....	32
3.5.3 Validation on Scan Cell Sequence.....	34
3.6 Summary of Chapter.....	35

Chapter 4 Results and Discussions.....36

4.1 Generation of Simulation Raw File and Mask File.....	37
4.1.1 Modification for Format of Scan Signal.....	37
4.1.2 Separation of Clock Cycle.....	38
4.1.3 Format of Simulation Raw File and Mask File.....	38
4.2 “Differentiate and Display” Feature.....	40
4.2.1 Differentiation of Two Raw Files.....	39
4.2.2 “Scan Compare” as Result from “Differentiate and Display” Feature.....	41
4.3 Masking Bogus 'x' Signal.....	43
4.3.1 Generation of Mask File.....	44

4.3.2 Collection of Silicon Data from Healthy Unit before Implementation of 'X' Filter.....	43
4.3.3 Collection of Silicon data from Healthy Unit after Implementation of 'X' Filter.....	45
4.3.4 Avoid Over Masking of Valid Scan Signals.....	46
4.4 Validation Tool.....	48
4.4.1 Extraction of Simulation Data.....	48
4.4.2 Collection of Silicon Data from Healthy Unit.....	49
4.4.3 Generation of Invert Database and Golden Simulation Raw File.....	50
4.4.4 Validation Result.....	52
4.5 Impact of Implementation on Real Case.....	54
4.5.1 Fault Isolation without Developed Tool.....	54
4.5.1.1 Signal Tracing and Determination of Upper Boundary.....	55
4.5.2 Failure Analysis based on Fault Isolation without Developed Tool.....	56
4.5.3 Fault Isolation after the Integration of Developed Tool.....	57
4.5.3.1 Determination of Upper Boundary after the Integration of Developed Tool.....	58
4.5.3.2 Signal Tracing and Determination of Upper Boundary.....	59
4.5.4 Usage of Validation Tool on Scan Signals in the Boundary Formed.....	61
4.5.5 Failure Analysis based on Fault Isolation with Integrated Tool.....	62
4.5.5.1 Unit Thinning and Signal Probing.....	64
4.5.5.2 Unit Preparation and Transmission Electron Microscopy (TEM).....	66
4.6 Discussion and Summary of Chapter.....	67

Chapter 5 Conclusion.....	68
References.....	69
Appendix A-D.....	74

LIST OF FIGURES

Figure 1.1 JTAG Structure.....	2
Figure 1.2 Scan Failure according to Fault Location.....	3
Figure 2.1 Graph of Transistor Size versus Year from year 1950.....	7
Figure 2.2 Image of Unit Cracking.....	8
Figure 2.3 The Role of Test and Failure Analysis in Product Development Cycle.....	9
Figure 2.4 The Basic Operation of BIST.....	10
Figure 2.5 Scan Cell formed by Latches.....	10
Figure 2.6 The Standard Flow of Failure Analysis Procedure.....	11
Figure 2.7 The Overall Fault Isolation Process.....	13
Figure 2.8 The Illustration of Suggested Symbolic X-Propagation Checking [9][11].....	15
Figure 2.9 Adapted Design Required to Handle Don't Care Situation of Registers [34].....	15
Figure 3.1 Flow Chart on Methodology of this Thesis.....	19
Figure 3.2 Block Diagram and Tester Setup in order to Collect Silicon Data.....	20
Figure 3.3 Procedure on Generation of Simulation Raw File.....	21
Figure 3.4 Flow of events in “Differentiate and Display” feature.....	25
Figure 3.5 Script to coordinate and map the bit location with their respective scan signal according to the mapping file from designer.....	26
Figure 3.6 Script to differentiate the value of interested signal, handle the invert characteristic as well as provide list of all mismatching signals	27
Figure 3.7 Operation of “Differentiate and Display” Feature in Different Mode.....	30

Figure 4.1 Command Used to Extract the Data in Waveform Viewer Application.....	37
Figure 4.2 Format of Mapping File before Modification.....	37
Figure 4.3 Format of Mapping File after Modification (Difference is Circled).....	37
Figure 4.4 Part of Scan Signals in Waveform Viewer Software Application.....	39
Figure 4.5 Extracted Data of Scan Signals from Waveform Viewer Application.....	39
Figure 4.6 (a) Mask File and (b) Raw File based on Extracted Data.....	39
Figure 4.7 The GUI for “Differentiate and Display” Feature.....	40
Figure 4.8 Result from “Differentiate and Display” Formed by Two Matching Raw File.....	41
Figure 4.9 Result from “Differentiate and Display” Formed by Two Mismatching Raw File.....	41
Figure 4.10 Result from “Scan Compare” Formed by Two Mismatching Raw Files.....	42
Figure 4.11 Scan Signal A (circled) Carrying Value ‘x’ in Waveform Viewer Application.....	43
Figure 4.12 Scan Signal Carrying Value ‘x’ in Extracted Simulation Data.....	43
Figure 4.13 Mask File Generated based on Extracted Data.....	44
Figure 4.14 Silicon Data Collected from “core 0” (a) and “core 1” (b) in Healthy Unit before the Implementation of Masking. Bit Location of Signal A is circled.....	44
Figure 4.15 Result from “Differentiate and Display” Showing Mismatch (as circled) between Signal that has ‘x’ Value in Simulation before the Implementation of Masking.....	45
Figure 4.16 Silicon Data Collected from Healthy Unit after the Implementation of Masking. Bit Location of Signal A is circled. (a) “core 0” and (b) “core 1”	46
Figure 4.17 Result from “Differentiate and Display” Showing Match between Signal that has ‘x’ Value in Simulation after the Implementation of Masking.....	46
Figure 4.18 Result from “Differentiate and Display” Showing Match between Signal that has Valid Value in Simulation after the Implementation of Masking.....	47

Figure 4.19 Mapping File from Designer with Stated Invert Characteristic (as circled).....	48
Figure 4.20 Simulation Data in Waveform Viewer Application.....	48
Figure 4.21 Simulation Data Extracted from Waveform Viewer Software Application.....	49
Figure 4.22 Silicon Data Collected from Actual Healthy Microprocessor Unit.....	49
Figure 4.23 Invert Database.....	50
Figure 4.24 Mask File Generated.....	50
Figure 4.25 Original Raw File from Simulation Data after Changing ‘x’ to ‘1’.....	51
Figure 4.26 Modified Raw File for Simulation Data after X-OR Operation with Invert Database for Every Clock Cycle.....	51
Figure 4.27 ‘Scan Compare’ Generated by “Differentiate and Display” Feature Shows that No Mismatch is Found between Silicon data and Simulation Data for this part of Mapping file in the Interested Range of Clock Cycle.....	52
Figure 4.28 The Identified First Failing Scan Signal in Increasing Clock Cycle.....	54
Figure 4.29 Signal A (as circled) has 8 Bit of ‘x’ in the Interested Clock Cycle.....	55
Figure 4.30 The Fault Isolation Process with Absence of Developed Tool.....	55
Figure 4.31 Signal B is having same Value as Signal A and it Matches between Failing Core and Reference Core.....	56
Figure 4.32 Result from Optical Test with Focus on Formed Boundary within the Sample.....	57
Figure 4.33 Signal A has Matching Value between Failing Core and Reference Core after the Implementation of Developed Tool.....	58
Figure 4.34 Signal C (as circled) has Mismatch Value between Failing Core and Reference Core in Clock Cycle 6bd9.....	59
Figure 4.35 The value of Signal C (as circled) in the Range of Interested Clock Cycle in Waveform Viewer Software Application.....	59
Figure 4.36 The Fault Isolation Process with Presence of Developed Tool.....	60
Figure 4.37 Signal D (as circled) is having Matching Value between Cores.....	61
Figure 4.38 “Scan Compare” File in Validation Mode for Interested Range of Clock...	61

Figure 4.39 The result from Optical Test that Shows the Abnormal Emission of Photon in the Suspected Failing Boundary.....	62
Figure 4.40 The Latest Failing Boundary after Elimination of Non-Suspected Device..	63
Figure 4.41 Graph of Stage Current vs Probe Voltage for the Output of Multiplexer....	64
Figure 4.42 Graph of Stage Current vs Probe Voltage for the Reference.....	65
Figure 4.43 The Illustration Image of Cross Surface of the Sample which Shows that there is Physical Contact between the Via of Output for Multiplexer and the Metal Layer of Voltage Supply.....	66

LIST OF TABLES

Table 3.1 Truth Table of the X-OR Operation between Invert Database and Signal's State.....	23
Table 3.2 Truth Table of AND-Operation between Mask File and Signal's State.....	29
Table 3.3 Result of Comparison after Masking of Silicon Data.....	29
Table 3.4 Result of Comparison for Scan Signals having 'x' in Simulation and Different Power-Up State before the Implementation of Masking.....	31
Table 3.5 Result of Comparison for Scan Signals having 'x' in Simulation and Different Power-Up State after the Implementation of Masking.....	32
Table 3.6 Comparison of Scan Signals having Value '0' from Healthy Cores after then Implementation of Masking.....	32
Table 3.7 Comparison of Scan Signals having Value '0' and '1' from Health Core and Failing Core Respectively after the Implementation of Masking.....	33
Table 3.8 Comparison of Scan Signals having Value '1' from Healthy Cores after then Implementation of Masking.....	33
Table 3.9 Comparison of Scan Signals having Value '1' and '0' from Healthy Core and Failing Core Respectively after the Implementation of Masking.....	33
Table 3.10 Data Collection and Analysis for Cores that has Mismatch between Simulation Data and Silicon Data.....	34
Table 4.1 Separation of Simulation Run Based on Clock Cycle due to Limitation From Waveform Viewer Software Application.....	38
Table 4.2 Result From the Validation Tool for Scan Out and All Clusters From Interleaved Scan.....	52

LIST OF EQUATIONS

Equation 3.1 Formula to Calculate the Mismatching Percentage between Silicon Data Collected and the Simulation Data.....	35
---	----

CHAPTER 1

INTRODUCTION

1.1 Background

Yield is defined as the ratio of microprocessor units in a wafer that can perform and function as expected to the sum of microprocessor units produced in a wafer. It is mainly affected by the fabrication process and it carries huge commercial impact on mass production [1]. As the size of transistor shrinks from time to time, fabrication process faces challenge in maintaining the manufacturing yield. In order to minimize the rate of defect, fabrication process is improved and fixed continuously with data and root cause from defect mechanism [2]. The process and methodology developed to identify the root cause is named failure analysis. Failure analysis is crucial in product development cycle, it can benefit the wafer fabrication process from the production of first silicon to the package development. Of all the failure analysis steps, fault isolation is often the first step in order to define the failing boundary within the microprocessor unit, the location where the defect is expected [3]. Fault isolation relies heavily on the Design for Testability (DFT) and Design for Debug (DFD) or most of the times are combined and named as DFx feature on the chip. DFx is a design technique that includes the testability feature at architectural, layout and circuit level so that test application and diagnostics can be applied by debug engineer [4].

One of the most popular DFx method is by implementing the scan capability, which recognized by Joint Test Action Group (JTAG), IEEE standard 1149.1-1990, entitled IEEE Standard Test Access Port (TAP) and Boundary-Scan Architecture. The

common standard consists of Test Access Port (TAP) controller, Scan Instruction Register and Scan Data Register within the device. The communication with external input is enabled with 5 pins to control the write and read of test instructions and data as well as their sequence, timing and priority with the architecture as shown in Figure 1.1 [5].

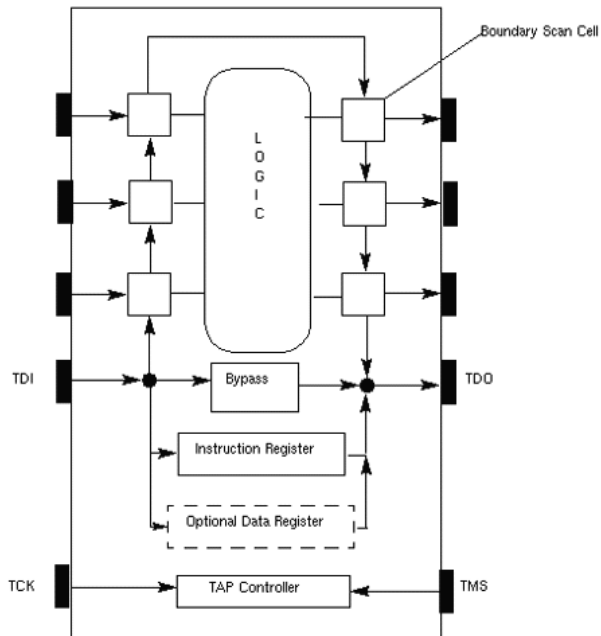


Figure 1.1: JTAG Structure

Pin Test-Data-In (TDI) and Test-Data-Out (TDO) are used to write and read the data respectively, while pin Test Clock (TCK) and Test Mode Select (TMS) are used to determine the timing and selection of either instruction register scan or data register scan. There is also an optional Test Reset (TRST) pin that can reset the test signal when it is necessary [6].

The main function of scan architecture in DfX is to provide visibility points along the circuit. Scan cell is formed by components of flip-flop or latch which able to store the

state of the circuit, while scan chain is formed by numbers of scan cells in a chain with capability to communicate with the combinational logic as shown in Figure 1.2 [7].

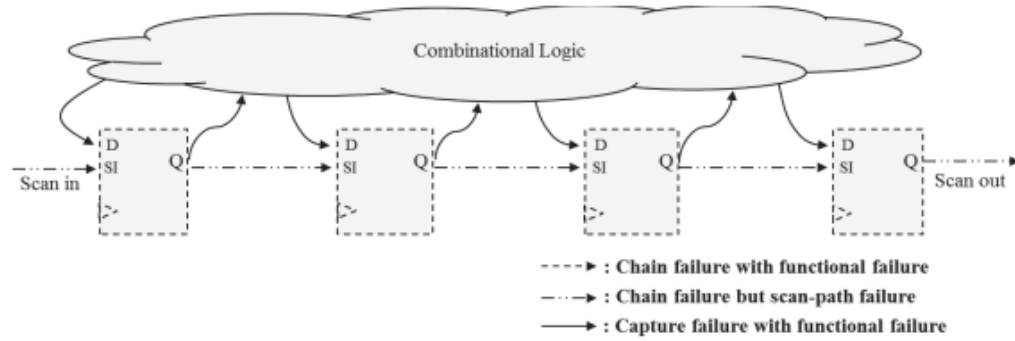


Figure 1.2: Scan Failure according to Fault Location

With the flexibility to apply input data into the scan chain and the combinational logic, test result can be analyzed by matching with the expected output data. In situation of output data mismatch with the expected data, failing boundary containing defect will be located and formed, whether it is within the scan cell chain or combinational logic, depends on the type of tests applied. Signal tracing with the help of simulation will further isolate the failing boundary, improve the debug process with higher effectiveness [8].

However, there are drawbacks and limitations on the scan architecture that might mislead the debug process, which indirectly reduce the yield and causes negative impact to the production cost. It is observed that scan cell sequence within the scan cell chain is important to match with the simulation platform in order to locate the failing boundary accurately. Besides, there are also presence of bogus signal which carries the ‘don’t care’ condition in simulation, floating to the power-on state.

1.2 Problem Statement

In order to isolate the failing boundary effectively and precisely, the presence of visibility points along the circuit and their accuracy are vital to the debug process. However, the state of bogus signal which carries ‘don’t care’ situation in simulation alters the validity of the visibility point and might mislead the debug process [9]-[12]. Since the bogus signal floats to power-on state, comparison data might show mismatch when we compare the healthy core and core that contains defect. The mismatching data might mislead the debug engineer to form failing boundary based on invalid test result.

Apart from that, the concern of the scan architecture is that the current validation for correlation between silicon data and simulation data is performed manually with eye-ball validation which has low effectiveness and low coverage along the scan cell chain [13]-[16]. In case the scan sequence within the scan cell chain differs with simulation or expected sequence, it will mislead the debug process and the failing boundary formed might not be valid and accurate which further leads to lower success rate of root causing the failing mechanism. Thus, a tool that can utilize the available simulation data along the period of test with minimum 90% of correlation with silicon data as well as capability to recognize and filter the ‘don’t care’ signal in diagnosis will help to improve the quality of fault isolation, and directly improve the yield, quality and cost effectiveness of the product fabrication [14]-[17].

1.3 Research Objectives

The research objectives for this dissertation projects are defined as:

- i. To enhance the fault isolation process by integration of bogus signal filter in the silicon data collection process. All the bogus signals presented in simulation data should be masked and filtered by the integrated tool to avoid the situation where the invalid data mislead the debug process.
- ii. To develop a validation tool that will automate the intensive validation process between simulation data and silicon data with coverage of minimum 90% of overall scan cells.

1.4 Research Scope

The research scopes for this thesis are defined as:

- i. Silicon data and simulation data from actual industry microprocessor, an Intel 6th generation core microarchitecture, Skylake with 14nm process technology will be imported for the algorithm design and development, proof of concept as well as the presentation of result and discussion in this thesis.
- ii. The methodology used for scan chain data collection as well as the test pattern generation will only be discussed briefly in this thesis.
- iii. The number of scan cells and the details in scan architecture will not be further discussed as they are part of Intellectual Property (IP) owned by Intel.

1.5 Thesis Outline

In chapter 2, literature review on thesis topic is presented. The reasons behind failure analysis and the DFX feature developed throughout the years are explained. Conventional fault isolation and failure analysis will be discussed so that the results in this thesis can be better understood. The chapter is completed by analysis of available researches and algorithm developed for similar purpose of this thesis in optimizing fault isolation from the aspect of scan architecture and sequences.

In chapter 3, methodology of this thesis will be presented. The procedure on logistics and data collection, followed by the design and development of features and core algorithm will then be presented. Details explanation in each features are presented. Proof of concept and the procedure in result analysis will then be explained in the last section of this chapter.

Chapter 4 is mainly about result and discussion where the instant output and result of this thesis will first be presented. The result from data collection and feature developed throughout this thesis will be presented. The difference in fault isolation process with and without this thesis will also be compared followed by the improvement of the overall failure analysis in root cause the defect mechanism with the help of this thesis. This chapter is completed with an actual failure analysis case from industry to emphasize the importance and impact of the thesis.

Last but not least, chapter 5 will discuss mainly about the conclusion and the future works of this thesis in future.

CHAPTER 2

LITERATURE REVIEW

This chapter of thesis discusses about the previous studies on related field including the reason behind failure analysis and DfX feature, followed by explanation on conventional fault isolation and failure analysis process. Previous works on handling bogus 'x' signal as well as post silicon scan cell validation. By study and analyzing on available knowledge and previous works, reference and benchmark can be made along with the latest technology information and knowledge.

In year 1965, more than half a century ago, “Cramming More Components onto Integrated Circuits” by Gordon E. Moore was published, which claims that the amount of transistors per square inch implemented in an integrated circuit will be doubled every year. Not only the statement is proven correct by the fact of history, it is even recognized as “Moore’s law” since then [18], [19], which the frequency of doubling was revised to every two years in year 1975 [20]. Figure 2.1 shows that the shrinking in size of transistors since the introduction of Moore’s law. With the decreasing size of transistors, the complexity and difficulty in fabrication process of microprocessor has increased, more physical defects are observed [1].

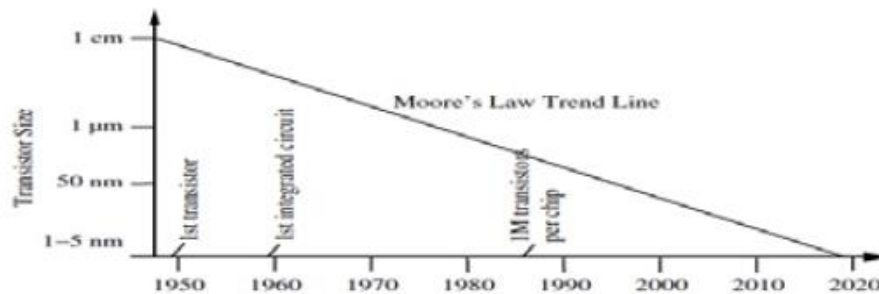


Figure 2.1 Graph of Transistor Size versus Year from year 1950

2.1 The Reasons Behind Failure Analysis

Studies show that if the physical defects are not well taken care, shrinking size of transistor will greatly affect the performance of microprocessor. Although the impacts are still negligible before the 350 nm-regime [26], statistic shows that the yield of microprocessor drops from 90% to approximately 50% entering the 90 nm process [27], and further dropped to approximately 30% with 45 nm process [28]. If the issue is continuously neglected, the physical defects on transistors can go as far as trading off the performance gained from specific generation of process technology [29].

Note that the microprocessor units are fabricated in the form of wafer. The High Volume Manufacturing (HVM) and engineering data show that the location of a unit of microprocessor in a wafer determine whether the unit is healthy or having defect [30]. Based on previous cases, there are several types of possible failures such as metal etch defects, via defects and unit cracking as shown in Figure 2.2 [3].

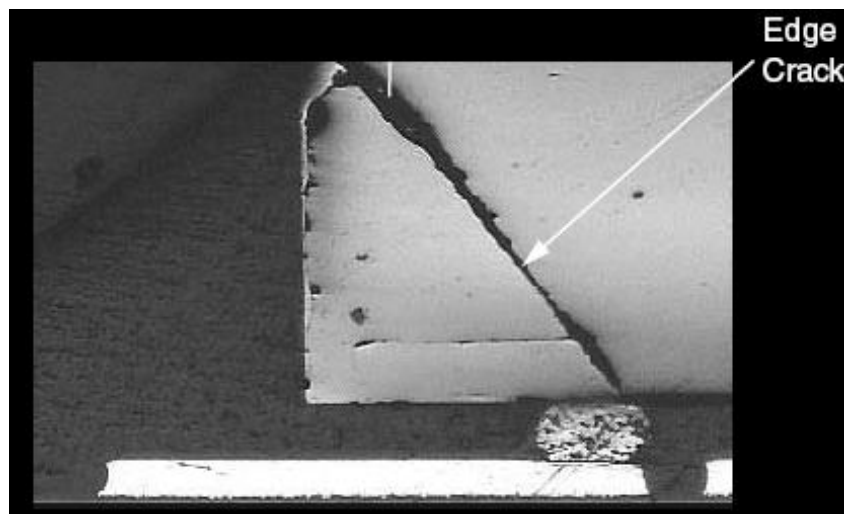


Figure 2.2 Image of Unit Cracking

The health of microprocessor in wafer is affected by the maturity of the fabrication process. By implementing failure analysis to root cause the defect mechanism, corrective

action can be installed on fabrication process on specific location in the wafer, Defect Per Millions (DPM) can be minimized, and the cost effectiveness can be improved [31].

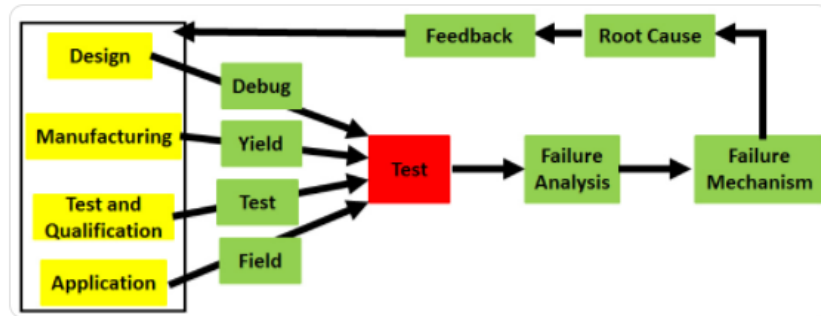


Figure 2.3 The Role of Test and Failure Analysis in Product Development Cycle

2.2 DFx Feature

In order to improve the yield performance of the fabrication process, DFx feature is implemented during the design process of microprocessor. DFx feature is defined as the inclusion of observability and controllability within the circuit in the design and product development cycle [4]. The main objective is to improve the effectiveness of debug process and the performance of the DFx feature will affect the quality of debug process as well as the result from failure analysis.

2.2.1 Build-In-Self-Test (BIST)

One of the most common DFx feature is the implementation of Build-In-Self-Test (BIST). There are three main components in the operation of BIST which are the test pattern generator, the test controller and the comparator. One of the advantage of BIST is that the test pattern will be generated within the circuit which help to save the cost of debugging. The pattern generated is applied to the circuit-under-test by the test controller

and the output response is compacted and compared with the reference signature from ROM [32].

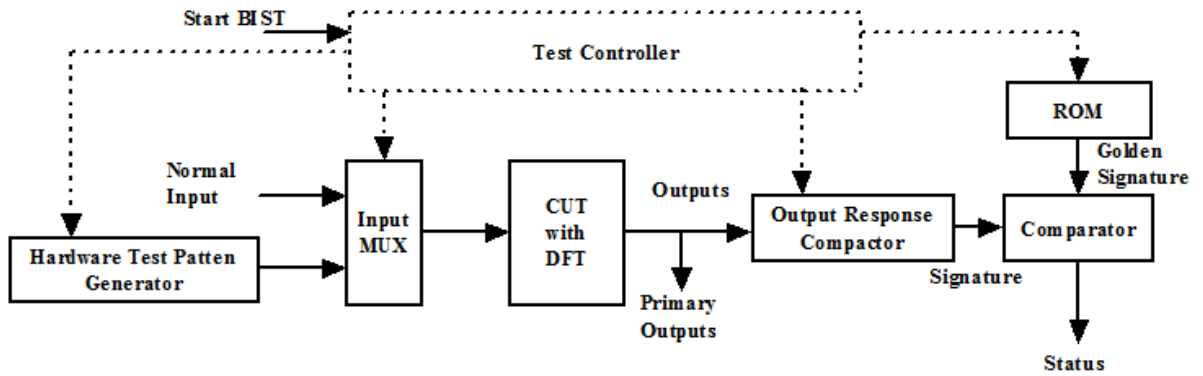


Figure 2.4 The Basic Operation of BIST

2.2.2 Scan Architecture

In this thesis, the DFX feature discussed is the architecture of IEEE 1149.1 boundary scan standard [33]. Boundary scan architecture consists of scan cells that help to capture and store the state of the signal which provide a visibility point of the signal within the circuit [34]. The scan cell layout that utilize two latches is presented in Figure 2.5 below.

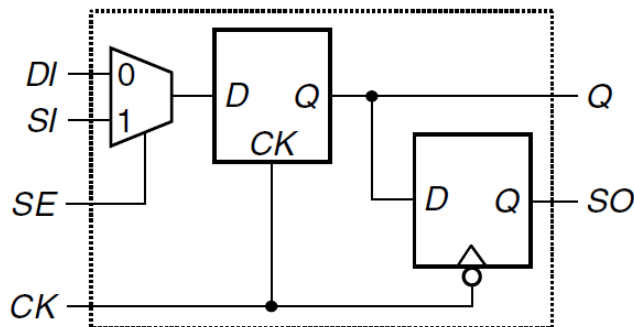


Figure 2.5 Scan Cell formed by Latches

2.3 Conventional Fault Isolation and Failure Analysis

In order to further familiarize and understand the impact of this thesis, the overall conventional fault isolation and failure analysis procedure is discussed and explained in details, however, due to Intellectual Property (IP) concern, some of the data and results are prohibited in this thesis report. Failure analysis is defined as the process of identifying physical defect and component failure of a microprocessor unit with the help from series of both electrical data and physical data analysis. Fault isolation, or sometimes termed as fault localization is part of failure analysis process which focuses on forming a suspected failing boundary within the electrical circuit with the help of the DfX features [36]. The standard flow of failure analysis procedure is presented as Figure 2.6 [31].

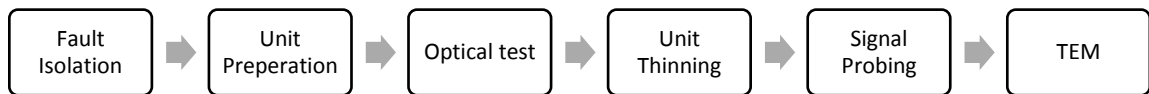


Figure 2.6 The Standard Flow of Failure Analysis Procedure

2.3.1 Fault Isolation

Fault isolation mainly involves electrical data collection and analysis. In multi-core design microprocessor, which the data is imported from in this project, the health of silicon data is determined by comparison between cores in a unit, instead of comparing with another healthy unit. This is to simplify and ensure that the tests can be executed online with the help of basic tester that can load only one unit. This algorithm helps to reduce the cost by avoiding the needs to obtain a complex tester that can execute tests on two units in parallel.

Failing core is identified when the final Linear Feedback Shift Register (LFSR) value of a core is different among the cores. After the identification of failing core, the first failing scan signal and its clock cycle are determined in order to form a failing boundary. Similar to the identification of failing core, the value of scan signals from failing core are compared with scan signals from healthy cores and the signal that has different value with scan signals from all the other cores is identified as the failing scan signals, the earliest clock cycle that contain any failing scan signal is recorded as first failing cycle. It is important to start the fault isolation with first failing signal and clock cycle as the consequence failing signal and clock cycle might not be valid as the mismatching data is caused by the driver from first failing signal and clock cycle in flow of circuit but not caused by the actual defect mechanism.

The first failing scan signal with its failing clock cycle which is also the first failing clock cycle is defined as the lower boundary of the fault isolation process. The process now requires the help of the simulation platform which the test patterns are loaded and presented. Starting from the lower boundary, the driver of signal is traced device by device until there is a scan cell or observability point in the circuit, the scan cell or observability point is compared with the exact signal from the healthy core, if the result matches with the signal from healthy cores, fault isolation process is completed by defining it as the upper boundary. If the result mismatches, further tracing is required until the scan signal from failing core matches with the same scan signal from the healthy cores [32].

In the end of fault isolation process, failing boundary containing a lower boundary and upper boundary within microprocessor is determined and formed, the defect

mechanism and component failure is expected within the boundary. The overall fault isolation process is presented and shown as Figure 3.3.

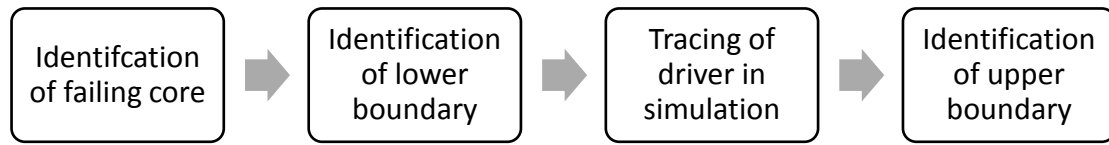


Figure 2.7 The Overall Fault Isolation Process

2.3.2 Failure Analysis

The failing microprocessor unit will now be transferred to be tested by optical tester in order to further minimize and isolate the failing boundary. Optical tester is used to capture emission of photon as result from series of combination and generation of electron-hole pairs within the transistor. Abnormal strong emission of photon can be observed should there is component failure in the interested boundary. Unit preparation is needed prior the optical test so that the infrared wave from tester can penetrate to the surface of interested layer on the microprocessor unit. The data from optical test is then be analyzed and the failing boundary is further isolated based on the abnormal emission of photon from the failing component.

The number of suspected device is reduced dramatically in this stage of failure analysis process and is suitable to go for probing due to low number of suspect candidates. Similar to the optical test, special surface preparation is needed so that the probe tip can be landed on the suspected component. After the analysis of optical data, the sample will then be thinned to the interested layer, probing will be carried out to collect the data related to possible failure of the component such as short circuit, open circuit and resistive path.

The result of probing is one of the last process before the defect is found and exposed physically with clear image.

The location of defect within the sample will be determined with the analysis and study on the probing data as well as the physical layout of the suspect candidates. The surface of the sample will then be thinned to the interested location and Transmission Electron Microscopy (TEM) will be used to capture the image of the physical defect in the microprocessor unit. The image of physical defect can be used to develop and improve the fabrication process, increasing the yield and maximize the cost effectiveness of the manufacturing process.

2.4 The Presence of ‘x’ State in Scan Signals

Due to the demands on smaller size and low power design of microprocessor, utilization of logic optimization and logic synthesis such as usage of don’t care ‘X’ condition is employed [9],[34]. The implementation of don’t care ‘X’ situation should be handled carefully as it might cause problems from the design, implementation to the verification process of the microprocessor [10]. There are numbers of solutions on handling of Don’t Care ‘X’ situation in the previous studies which most of them requires the involvement of early stage of product development cycle such as the architecture and design stage.

Few studies suggest the method to analyze and study the propagation of signal with don’t care ‘x’ value and minimize the number of initialized registers with symbolic x-propagation checking as shown in Figure 2.8 [9], [11], [34]. The authors then uses

similar approach with improvement on the computation cost with an algorithm that study only parts of the bogus signal named as heuristic algorithm [9], [10].

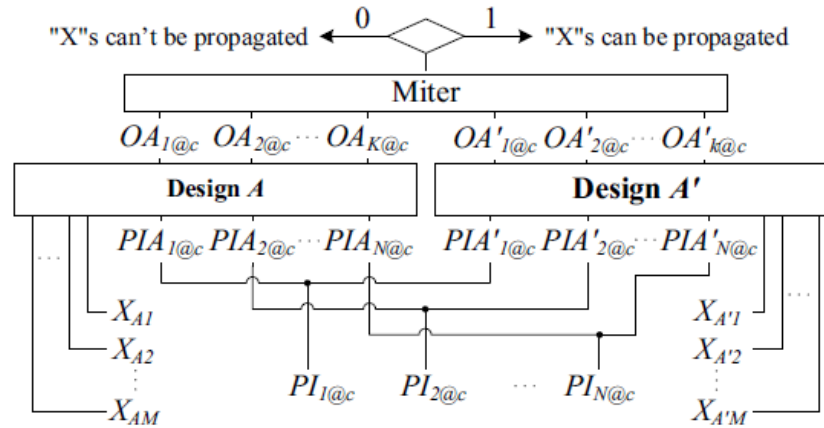


Figure 2.8 The Illustration of Suggested Symbolic X-Propagation Checking [9], [11]

The reduction of number of initialized registers acts as a feedback to the design and architecture stage. This solution of handling of don't care 'x' situation takes place only in the early stage of the product development flow as shown in Figure 2.9.

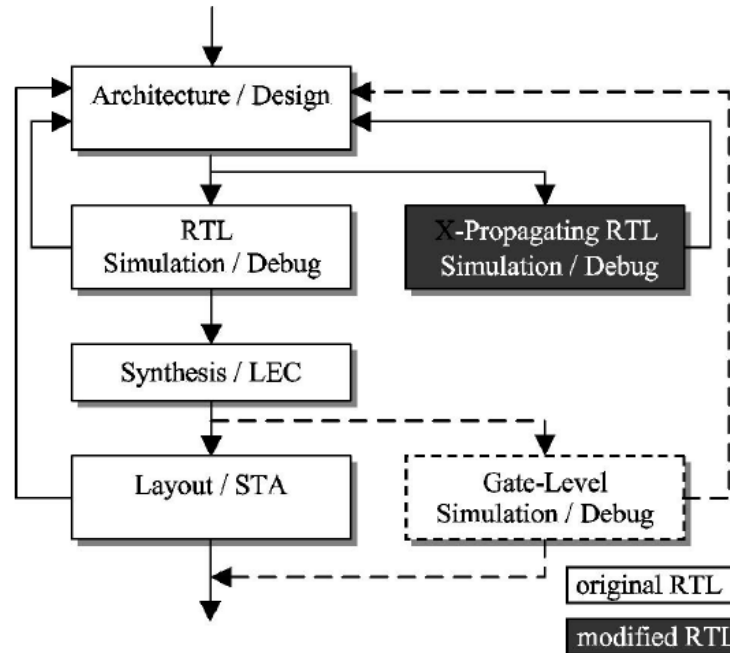


Figure 2.9 Adapted Design Required to Handle Don't Care Situation of Registers [34]

There is also suggestion to random replace the state of the signal with 'x' value to either value '1' or value '0'. The author consider only the functional performance of the microprocessor and hope for the best case scenario for the random assignment in the debug procedure [16]. However, some authors do not agree with this solution as they have experiments that show there might be hidden bugs in the DfX feature should the value of 'x' is replaced with '1' or '0' randomly [9], [34]. In this thesis, by using the simulation data collected, a post silicon algorithm is developed which not only minimize the usage of computation cost, but also cause minimum impact to the product development cycle.

2.5 Post Silicon Validation

Apart from the bogus signal filter, another objective of this thesis is to develop a post silicon validation tool for the scan chain architecture between the silicon data collected from the sample and the simulation data extracted from the test pattern generated. The correlation between silicon data and simulation data are important not only for fault isolation process, but also plays an important role in the development of automated fault isolation tool [35].

There are numbers of algorithm used to complete the fault isolation in post silicon validation from previous studies. One of the method studied is that a structural dependency graph that contain the representation of circuit in gate level can first be developed and the result of an approximate graph matching is used to correlate the data [13]. Besides, there is also method on feeding the bug data formed by comparison between architected states and simulation state into a machine for bug sites prediction where the machine is trained with huge database from previous experience [14]- [16]. Dynamic program slicing with

consistency checking is also another method to execute fault isolation in post silicon validation process [17].

2.6 Summary of Chapter

In order to understand the needs of failure analysis, brief history and background on evolution of transistor's size is presented. As the size of transistor decrease from time to time to achieve Moore's law, fabrication process faces challenges in maintaining the yield. Failure Analysis helps to rootcause the failure mechanism so that a solution can be applied in fabrication process in order to improve the yield performance.

The procedure in conventional fault isolation and failure analysis are studied and based on the few references in related field. By having the knowledge on the flow of fault isolation and failure analysis, the methodology and impact of this thesis are easier to be familiarized. Fault isolation helps to define the failing boundary within the circuit, where the physical defect and component failure are expected to be. The sample will then be sent to failure analysis, undergo the optical test and probing analysis before TEM image is taken to confirm the defect. As for the bogus signal handling, previous authors try to minimize the initialized register and use random replacement as solution. Structural dependency graph and bug site prediction are used to validate and correlate silicon data.

The studies of experience shared from the previous related works help to position and benchmark the objectives set in this thesis. The specification, advantages as well as the possible drawbacks from the related works provide the knowledge and vision in the process of algorithm development in this thesis.

CHAPTER 3

METHODOLOGY

This chapter discusses the methodology to complete this thesis including the necessary logistics and environment setup, the algorithm followed by the proof of concept, and the procedure of result analysis. Both simulation data and silicon data from Intel 6th generation microprocessor Skylake, a 14 nm manufacturing process technology unit are imported in this project in order to make sure that the thesis can be employed in industry. Programming language including Python and Unix are used in this thesis. The data are utilized not only to mask the bogus signal that are having power-up state, but will also be used to validate the sequence of scan cell and correlation between silicon data and simulation data.

A flow chart of methodology is presented in Figure 3.1 to provide an overview for this chapter. This thesis is started with the setting of problem statement and the objective of thesis, studies of previous works and knowledge can help to benchmark the objective set for this thesis. After the chapter of introduction is completed, logistic of thesis will first be prepared, “Differentiate and Display” feature will then be developed in order to process the simulation data or silicon data provided. The details of the feature will be explained in later part of the chapter. At the same time, simulation raw file as well as the mask file will be generated and the silicon data will be collected through the usage of tester. Result is collected and analyzed after the implementation of the algorithm, the result is assessed based on the impact on fault isolation and failure analysis process before and after the implementation of this thesis in the procedure.

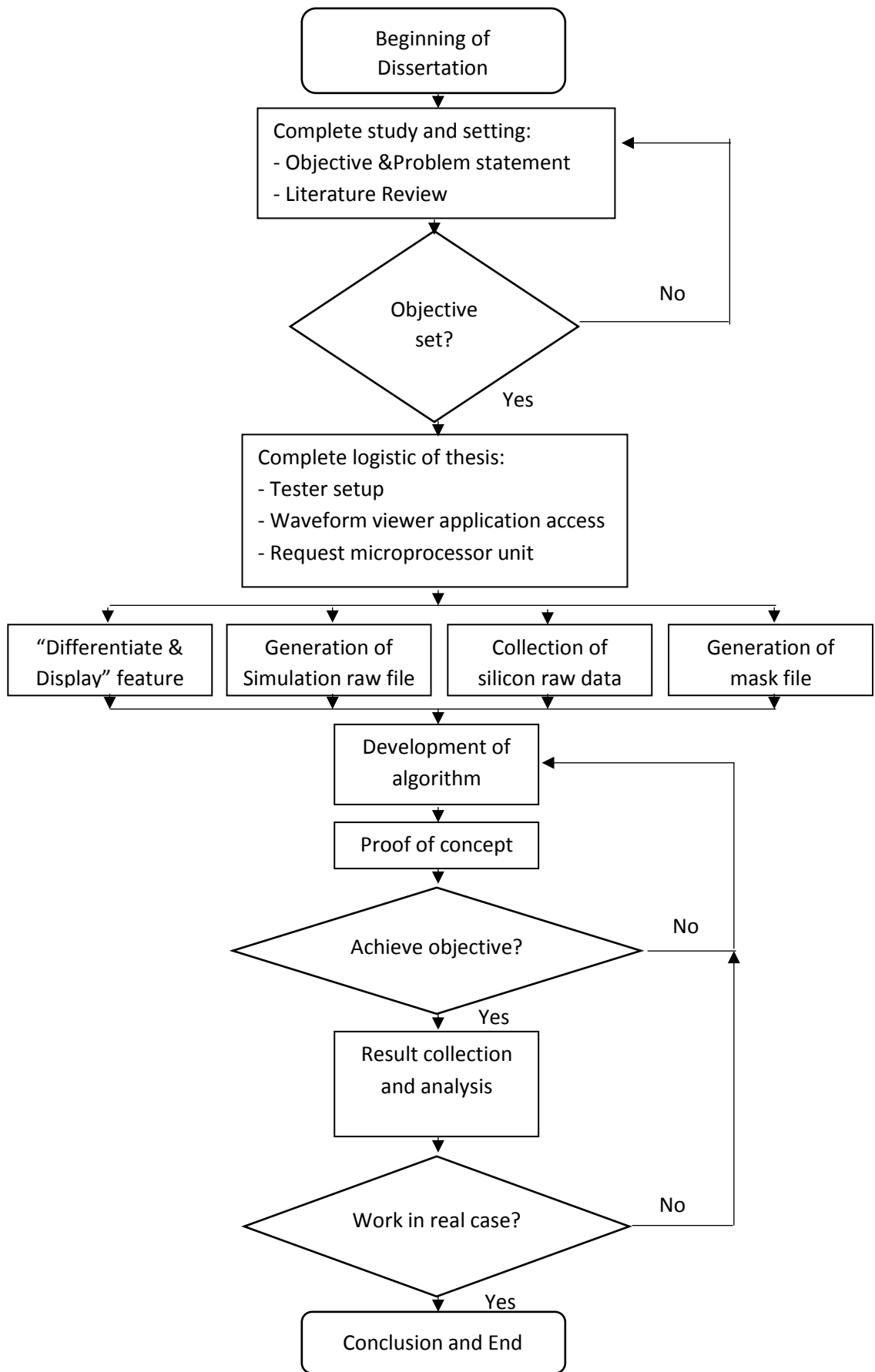


Figure 3.1 Flow Chart on Methodology of this Thesis

3.1 Setup of Tester and Software Applications in Fault Isolation

The tool and features developed in this thesis focus mainly on the improvement and development of the fault isolation process. The data from Intel 6th generation microprocessor, Skylake is imported for the usage of this project. The DfX features used in the sample is the JTAG standard pins. In order to provide input test pattern into the sample, a tester is needed. Standard tester involving the usage of a Field Programmable Gate Array (FPGA) and its own library is setup as shown in Figure 3.2.

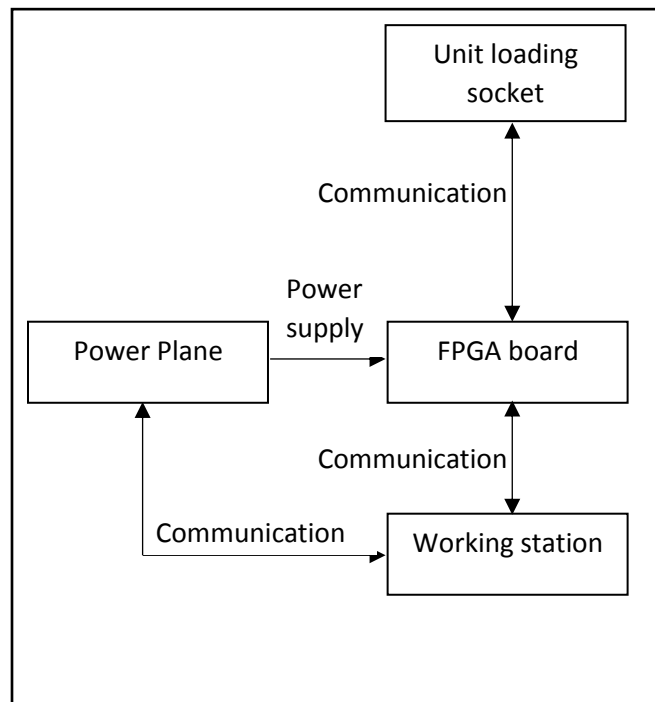


Figure 3.2 Block Diagram and Tester Setup in order to Collect Silicon Data

The tester is controlled by a Graphical User Interface (GUI) application connected to a standard working station. The interface platform provides flexibility to the user and developer to communicate with the microprocessor sample by utilization of the JTAG pins. The development of interface and communication with the microprocessor will not

be discussed in details as it is not the focus of this thesis. Apart from the communication, power supply to activate the sample throughout the failure isolation process is also provided by tester and connected power planes. With both the tester and power supply, silicon data is now ready to be collected.

As for the simulation data, they are displayed and presented by waveform viewer application. Test pattern is generated in Fast Signal Database (FSDB) format and a waveform viewer interface is needed in order to study and analyze the simulation of the test pattern. The function from waveform viewer interface is utilized to extract the simulation data including the signal name, their value as well as the clock cycle. The setup of both hardware tester and software application are vital to ensure that the data collection process is carried out in complete format.

3.2 Generation of Simulation Raw File

Data collection process is important to the application of this thesis especially in the validation mode. Since the silicon data is collected by using the tester, it is important that the collection of simulation data is discussed. Raw data is collected according to the test pattern, there should be a unique simulation data for every test pattern which includes the name of the scan signal and their state in every clock cycle throughout the test pattern.

An overview on the generation of simulation raw file is presented in flow chart below.

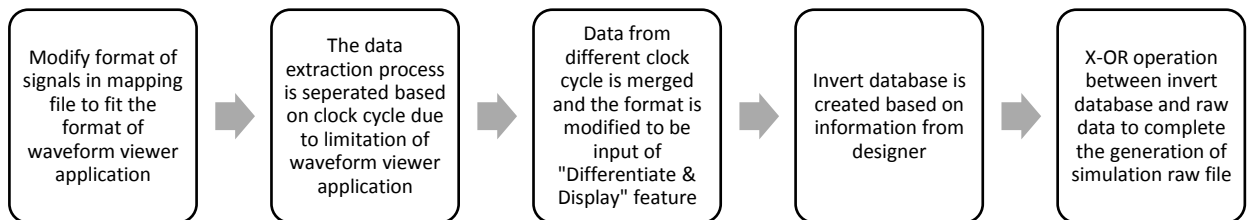


Figure 3.3 Procedure on Generation of Simulation Raw File

In this thesis, the simulation raw data is extracted by using the feature from the waveform viewer application. Command is called in Unix operating system installed in a work station in order to extract the state of each interested scan signal for each clock cycle throughout the test pattern. The name for some of the scan signals provided by the designer is not tally with the expectation from the waveform viewer application, modifications of the name of scan signals are required so that they can be recognized and the extraction of data can be processed. After the data is extracted from the simulation platform, it is necessary for the data to be displayed in the format that can be recognized by the GUI of tester. Simulation raw file is generated by rearranging the data which not only ensure that the data is ready to be utilized by the GUI of tester, but also help to minimize the usage of the disk space in storage since unnecessary information from the original data is removed.

3.2.1 Limitations and Solution in Raw Data Collection

There are two major roadblocks in the simulation raw data collection process which are the invert characteristic of some scan signals and the limitation of the waveform viewer application. In the design flow process of microprocessor, the strength of signal might become weak as it propagate from the driver to destination, inverter is added within the propagation path so that the signal is stable. Inverter will change the state of the signal to opposite status and this will cause miscorrelation in between silicon data and simulation data. In order to solve this roadblock, post process of both the collected silicon and simulation data together with the list of inverted signal from designer are utilized.

A list of invert database formed by all scan signals is generated with value '1' for inverted scan signal and value '0' for non-inverted scan signal. X-OR operation is

implemented between the invert database and the extracted simulation raw data. By using the X-OR operation, the state of inverted signal will be toggled and the state of non-inverted signal will be remained. The truth table for the scenario is shown in Table 3.1.

Table 3.1 Truth Table of the X-OR Operation between Invert Database and Signal's State

Original state	Characteristic	Invert database	After X-OR operation	Final state
0	Invert	1	1	Inverted
0	Non-invert	0	0	Remain
1	Invert	1	0	Inverted
1	Non-invert	0	1	Remain

Another roadblock of the generation of simulation raw file will be the limitation of the waveform viewer platform as it is not allowed to extract data more than 2000 clock cycles due to the size of the raw data file generated. The roadblock is solved by separating the generation process with interval of 2000 clock cycle for each iteration throughout the overall clock cycle of test pattern, followed by merging the data into single simulation raw data file.

3.3 'Differentiate and Display' Feature

This feature is important for the fault isolation or fault localization purpose, apart from being utilized in this thesis to define the health of scan signals in the circuit. Post processes of the data collected from both silicon and simulation are required so that the study and analysis of data are simplified. The silicon data collected from TDO pin is displayed in string of value '1' and '0', with their sequence formed by the sequence of the respective scan cells in the architecture. Different bits of same signal bus might not be

arranged properly and this create difficulty for fault isolation engineer to make any hypothesis based on the data. Besides, in the fault isolation process where only silicon data are collected, there is also issue miscorrelation between simulation data and silicon data caused by the presence of mentioned inverted scan signals in the microprocessor since the utilization of X-OR logic method is implemented only in simulation data.

Hence, it is important to design a “Differentiate and Display” feature which takes input of two raw data file, formed by two sets of silicon data from different cores for fault isolation purpose or pair of silicon data and simulation data for the validation purpose. The “Differentiate and Display” features is expected to handle the differentiation of data collected, define whether the pairs of input data is matching among themselves and their respective values in specific clock cycle. The feature is also expected to provide the list of mismatching scan signals for the usage of validation mode. Besides, there is a need to handle the toggling of inverted signals as well as arrangement of sequence based on the ascending order of bit for the same scan signal bus. Apart from that, extraction of certain range of clock cycle throughout the test pattern based on the input from user is also considered in the design of the feature. A series of features are planned and modified continuously as the “Differentiate and Display” feature is developed. The final design of the “Differentiate and Display” feature is presented in flow chart as shown in Figure 3.4. “Differentiate and Display” feature is scripted in Python programming language. Parts of the coding are extracted from the overall coding as shown in Figure 3.5 and Figure 3.6 so that each functions can be well explained. As for the complete coding of “Differentiate and Display” feature, it is available as reference in appendix of this thesis.